

```
node 4  
Run using Photon server on node 4
```

```
-s//4/dev/photon
```

```
Same as above
```

```
-x100 -y100 -h200 -w200
```

```
Run at initial position with initial
```

```
dimension of 200x
```

```
endif
```

```
endif
```

```
Standard
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <unistd.h>
```

```
/*
```

```
Local headers */
```

```
#include "ablibs.h"
```

```
#include "abimport.h"
```

```
#include "proto.h"
```

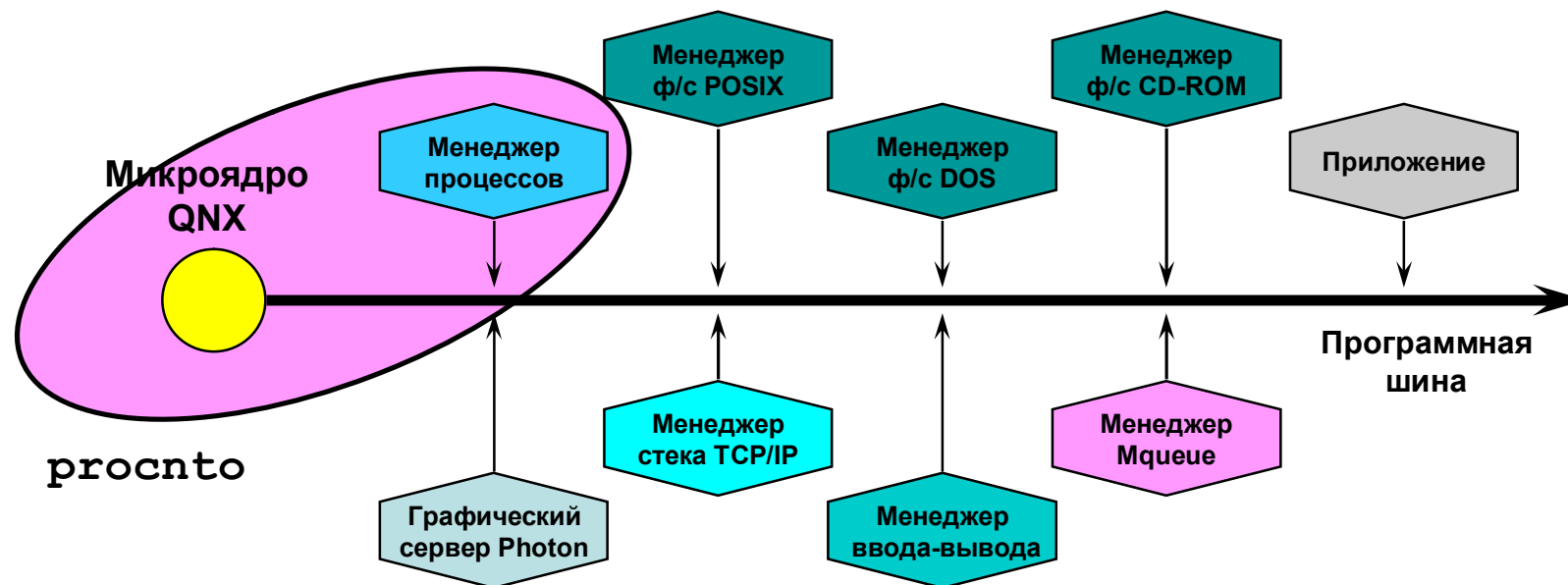
```
#include "abwidgets.h"
```

QNX Neutrino – штатные механизмы обеспечения отказоустойчивости систем жёсткого реального времени

ООО «СВД Встраиваемые Системы»

Сергей Зыль

Механизм №1 - Микроядерная архитектура

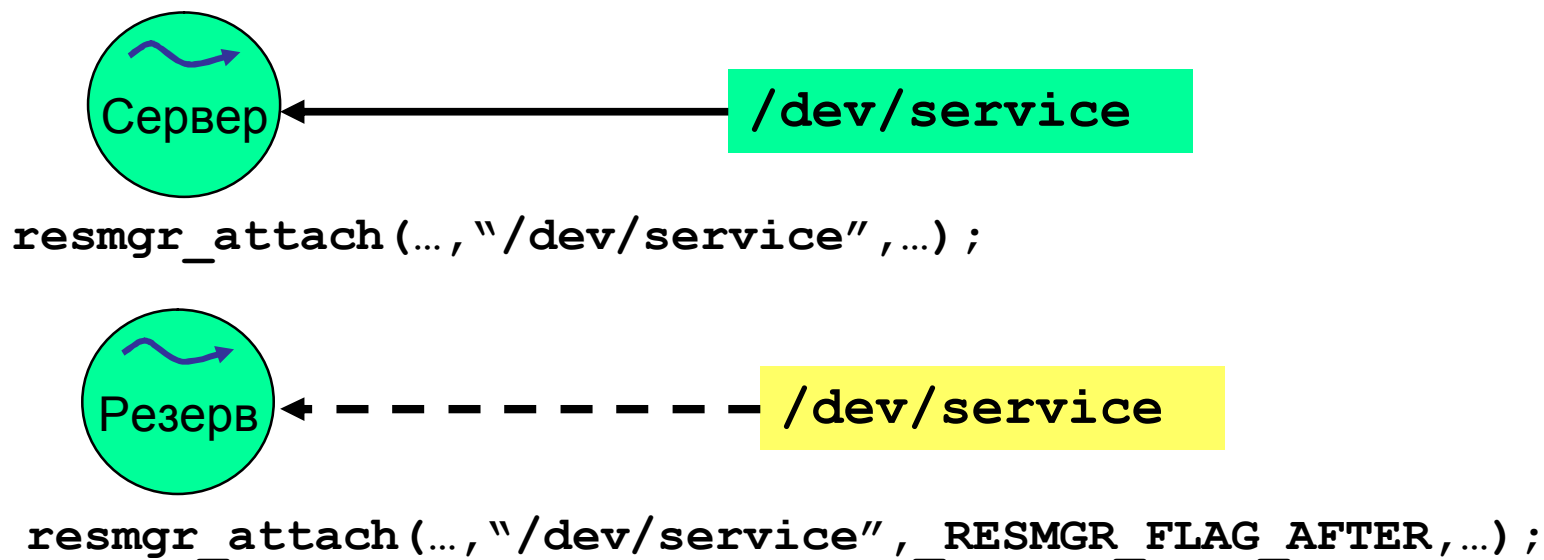


Т.е. драйвер QNX – прикладная программа!

- ✓ код ресурс-менеджера не расширяет SPoF системы (кроме функций-обработчиков прерываний)
- ✓ при разработке можно использовать все возможные инструменты отладки и анализа

Механизм №2 – Горячее резервирование сервисов

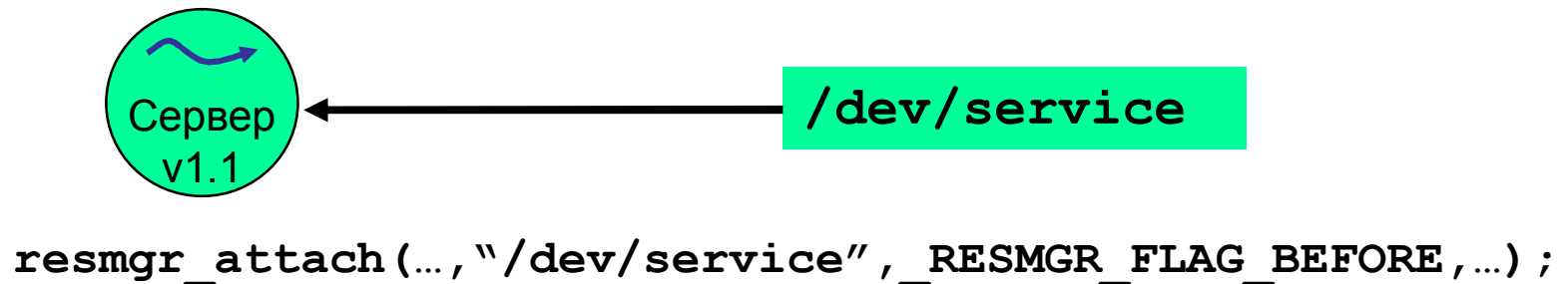
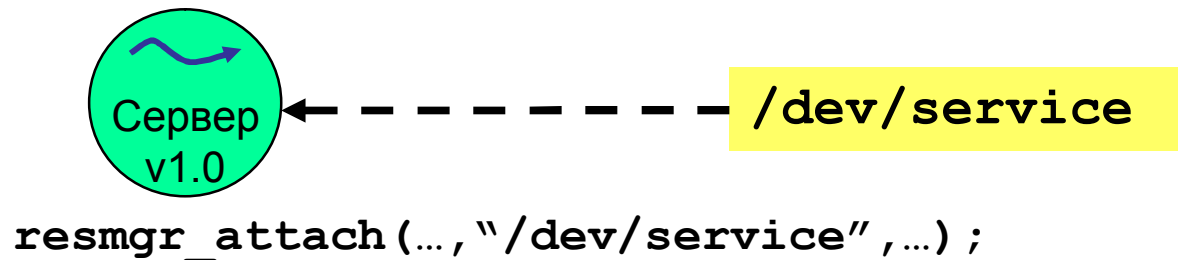
✓ Т.к. менеджер ресурса QNX прикладная программа, то можно запустить любое их количество!



Новый запрос `open("/dev/service", ...)` поступает на Сервер

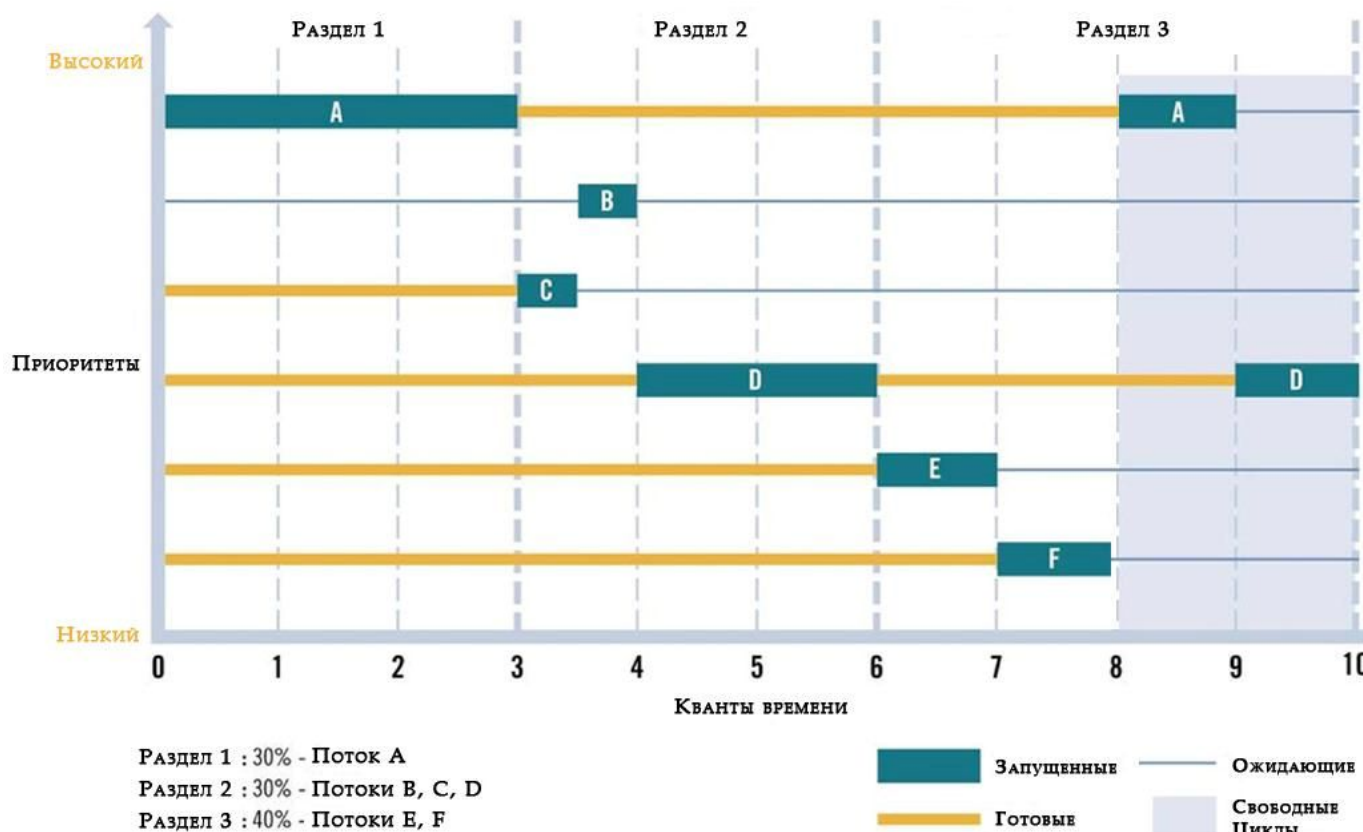
examples:
Run using Photon server on node 4
-s//4/dev/photon

Механизм №3 – Горячая замена активных сервисов



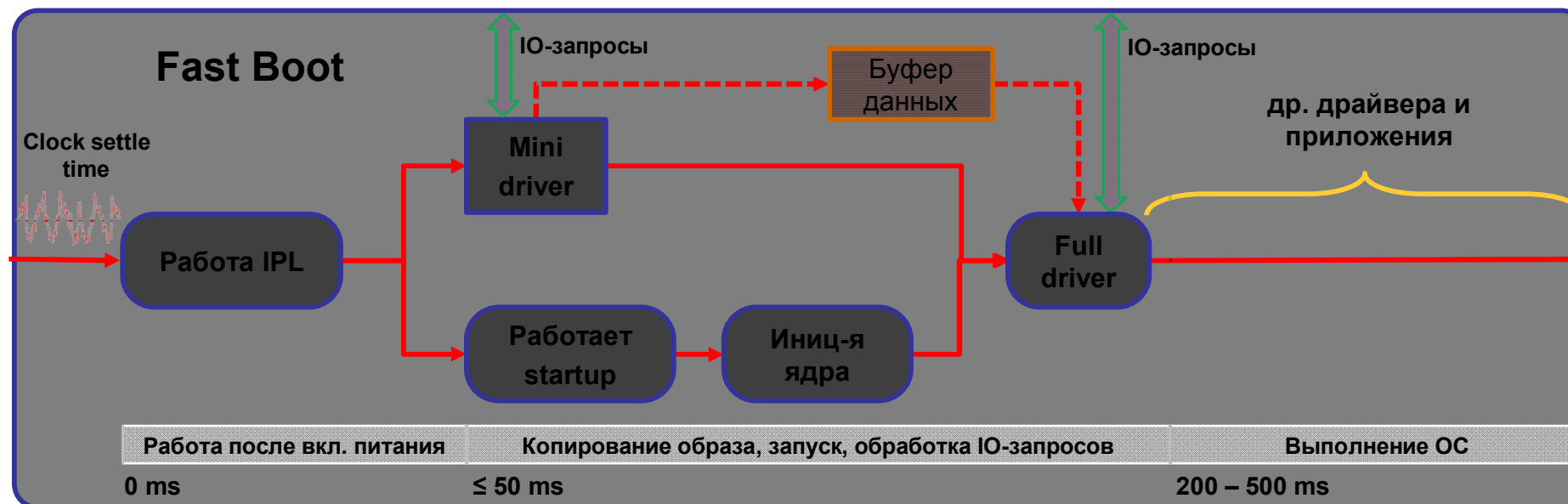
Новый запрос `open("/dev/service", ...)` поступает на Сервер v1.1

Механизм №4 – Адаптивное квотирование ресурсов



- ✓ Повышение защищённости и Кг АС за счёт невозможности монополизации ресурсов какой-либо программой (при DoS-атаках и некорректном коде);
- ✓ Сокращение трудозатрат на сопровождение АС на 25-30% (по оценкам QSS) за счёт локализации аномального поведения.

Механизм №5 – Быстрая активация устройств (IDA)



- ✓ Немедленный ответ на внешние запросы
- ✓ Продолжение или завершение работы прозрачным образом после того, как управление передаётся полной версии драйвера без задержек или потерь данных.

Механизм №6 – Отказоустойчивые файловые системы

В ОСРВ QNX Neutrino реализованы разнообразные файловые системы, позволяющие разработчикам систем реального времени решать широкий диапазон задач:

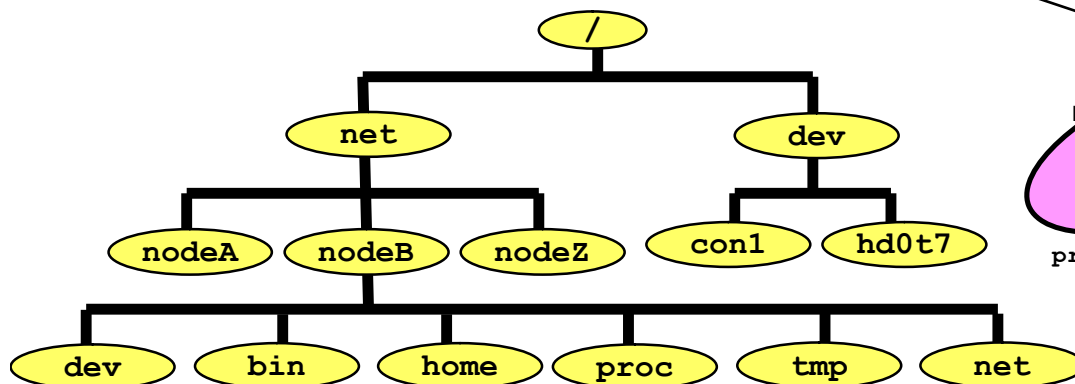
- ОЗУ-резидентные (Image, Shmem, io-blk, devb-ram)
- Для ППЗУ (FFS)
- Дисковые (QNX4, FAT32, NTFS, HFS+, ISO 9660, UDF)
- Сетевые (NFS, SMB)
- Виртуальные (inflator, пакетная)

Реализованы специальные отказоустойчивые ф/с:

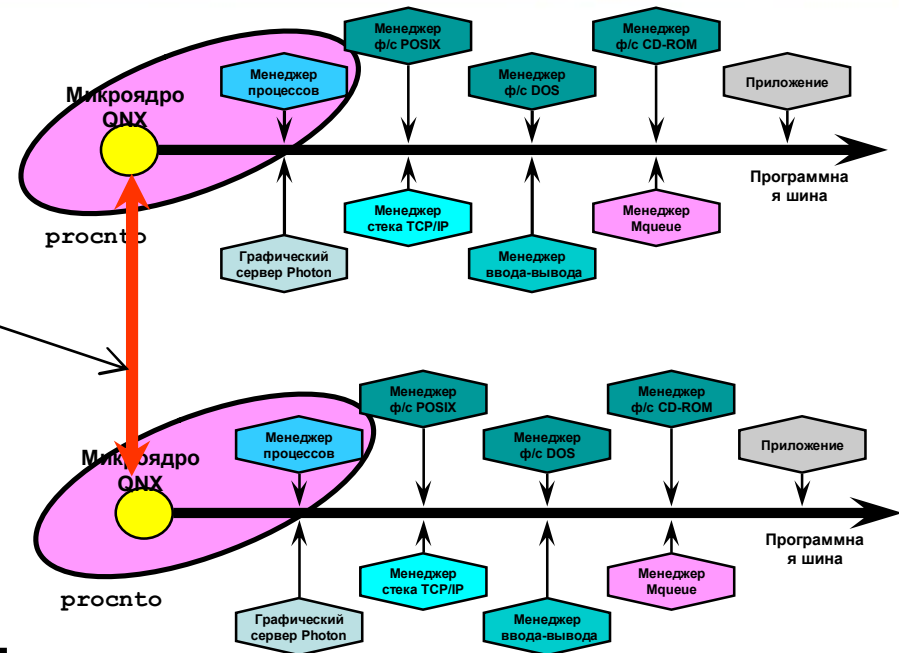
- ✓ **Power-Safe filesystem (QNX6)** – для жёстких дисков
- ✓ **Embedded transaction filesystem (ETFS)** – для Flash-памяти

Механизм №7 – Распределённая вычислительная среда

Технология QNX TDN на базе протокола
Qnet (4 уровень ISO OSI)



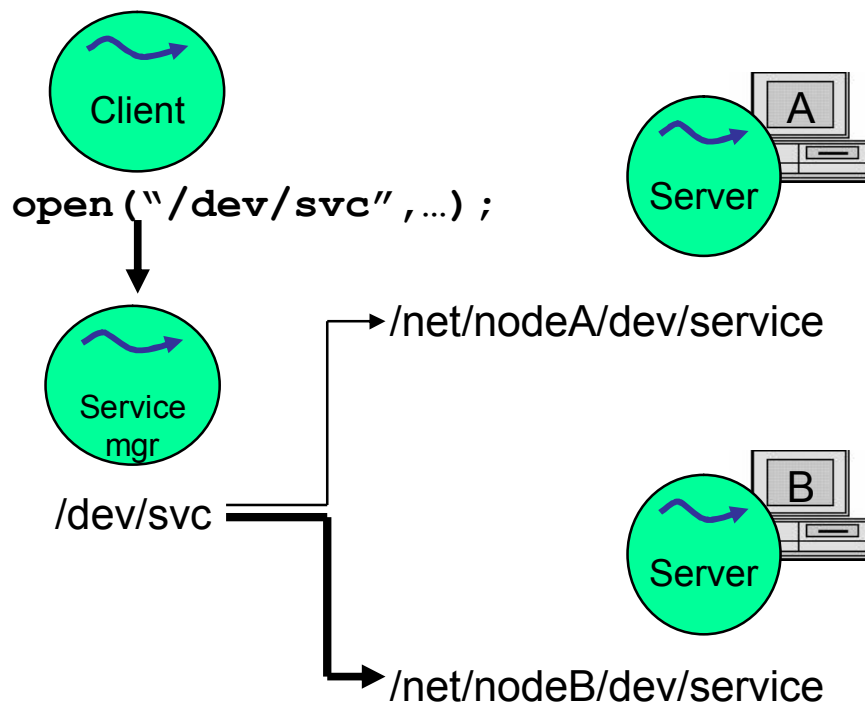
Менеджер Qnet регистрирует
префикс /net на своём узле



Запуск программы
в распределённом режиме:

```
on -n nodeB \
    -f nodeC \
    -t /net/nodeD~preferred:en1/dev/tty0 \
    /net/nodeE~exclusive:en2/bin/l
```


Механизм №8 – Распределение нагрузки между серверами

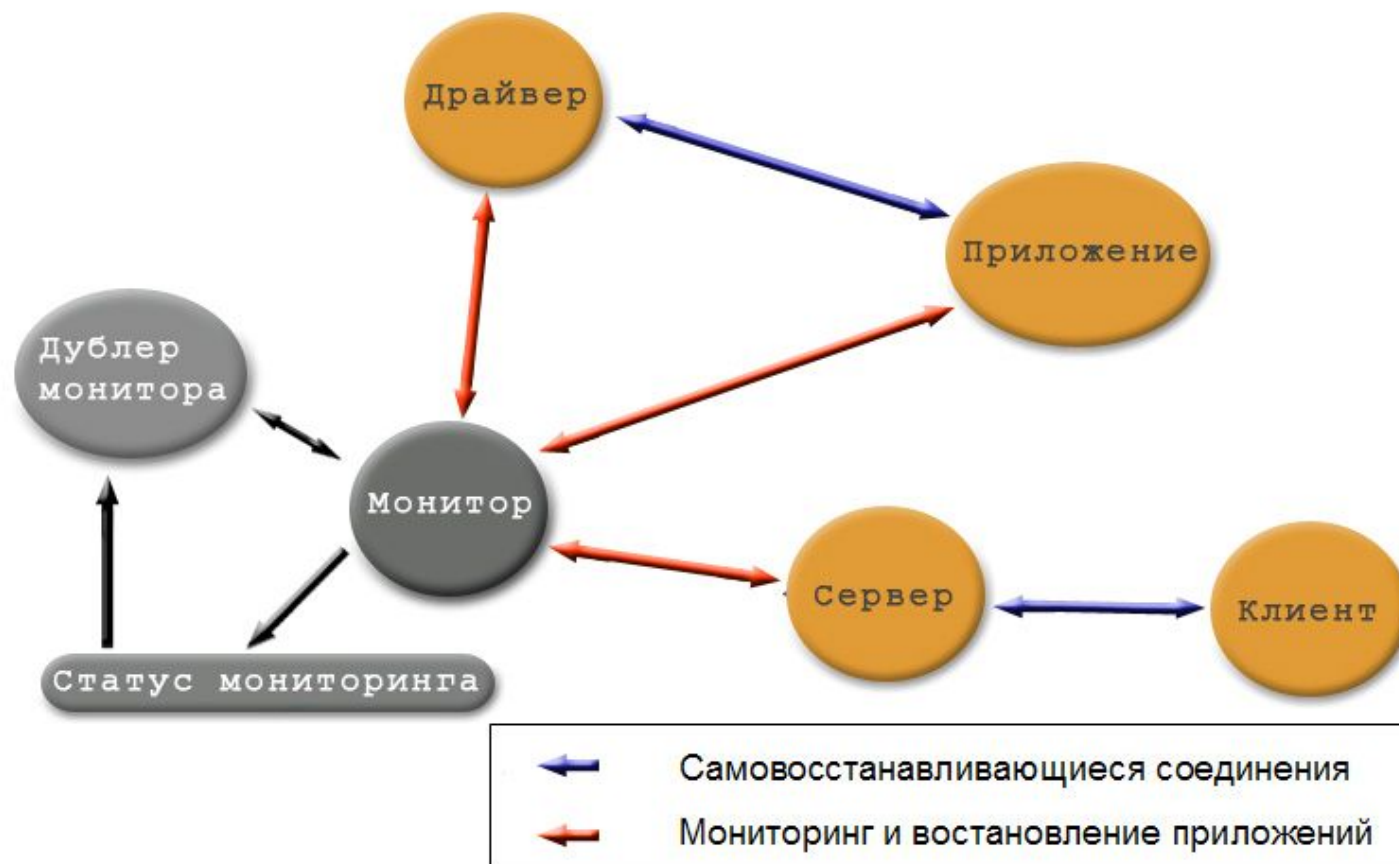


Код перенаправления клиентского запроса:

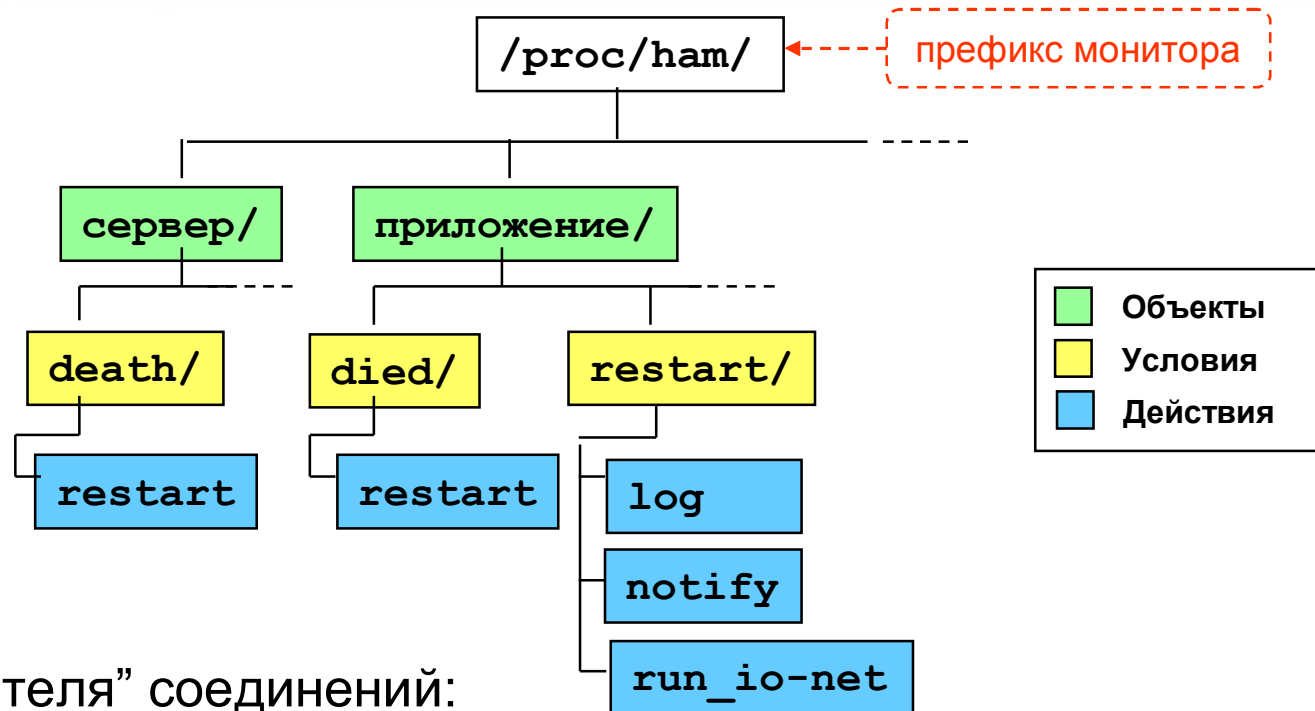
```
int
io_open (resmgr_context_t *ctp, io_open_t *msg,
         RESMGR_HANDLE_T *handle, void *extra)
{
    eflag = msg->connect.eflag;
    ftype = msg->connect.file_type;
    memset(&msg->link_reply, 0, sizeof(msg->link_reply));
    _IO_SET_CONNECT_RET(ctp, _IO_CONNECT_RET_LINK);
    msg->link_reply.file_type = ftype;
    msg->link_reply.eflag = eflag;
    msg->link_reply.nentries = 0;
    msg->link_reply.path_len = strlen(new_path) + 1;
    strcpy(((char *)msg + sizeof(msg->link_reply)), new_path);
    retlen = sizeof(msg->link_reply) +
        msg->link_reply.path_len;
    return _RESMGR_PTR(ctp, msg, retlen);
}
```

Новый запрос `open ("/dev/svc", ...)` поступит на один из узлов

Механизм №9 – Автоматизация восстановления процессов и соединений



Статус мониторинга



Код "восстановителя" соединений:

```
fd = ha_open( "/dev/device", O_RDONLY, recover_read_fd, "/dev/device", 0 );
...
int recover_read_fd( int old_fd, void *hdl ) {
    fname = (char *) hdl;
    delay(100); // allow server time to restart
    new_fd = ha_reopen( old_fd, fname, O_RDONLY );
    return new_fd;
}
```

```
amples:  
-s4  
Run using Photon Server on node 4  
-s//4/dev/photon
```

Спасибо за внимание!

ВСЕ рассмотренные механизмы
реализуются штатными средствами
дистрибутива QNX SDP 6.4.1

ООО «СВД Встраиваемые Системы»

Центральный офис:

196066, г. Санкт-Петербург,
Московский проспект, д. 212 А

тел.: (812) 373-41-17

факс: (812) 373-19-07

Технический офис:

191014, г. Санкт-Петербург,
ул. Госпитальная, д.3

тел./факс: (812) 578-02-45